

Subversion

Τριγάζη Ελισάβετ

Επιβλέπων: Μηνάς Δασυγένης –
<http://arch.ict.e.uowm.gr>

Το Πρόβλημα

Τα προγράμματα λογισμικού μπορεί να είναι μεγάλα και πολύπλοκα.

- Μπορεί να εμπλέκουν πολλά άτομα
- Μπορεί να απαιτείται συντήρηση των διαφόρων συναφών εκδόσεων
- Μπορεί να απαιτείται πρωτοτυποποίηση πιθανών χαρακτηριστικών, διατηρώντας παράλληλα τα υφιστάμενα χαρακτηριστικά

Η Λύση

Τα εργαλεία VCS (version control systems) προσπαθούν να αντιμετωπίσουν αυτά τα προβλήματα επιτρέποντας τη συντήρηση και αρχειοθέτηση των πολλαπλών εκδόσεων ενός τμήματος του λογισμικού:

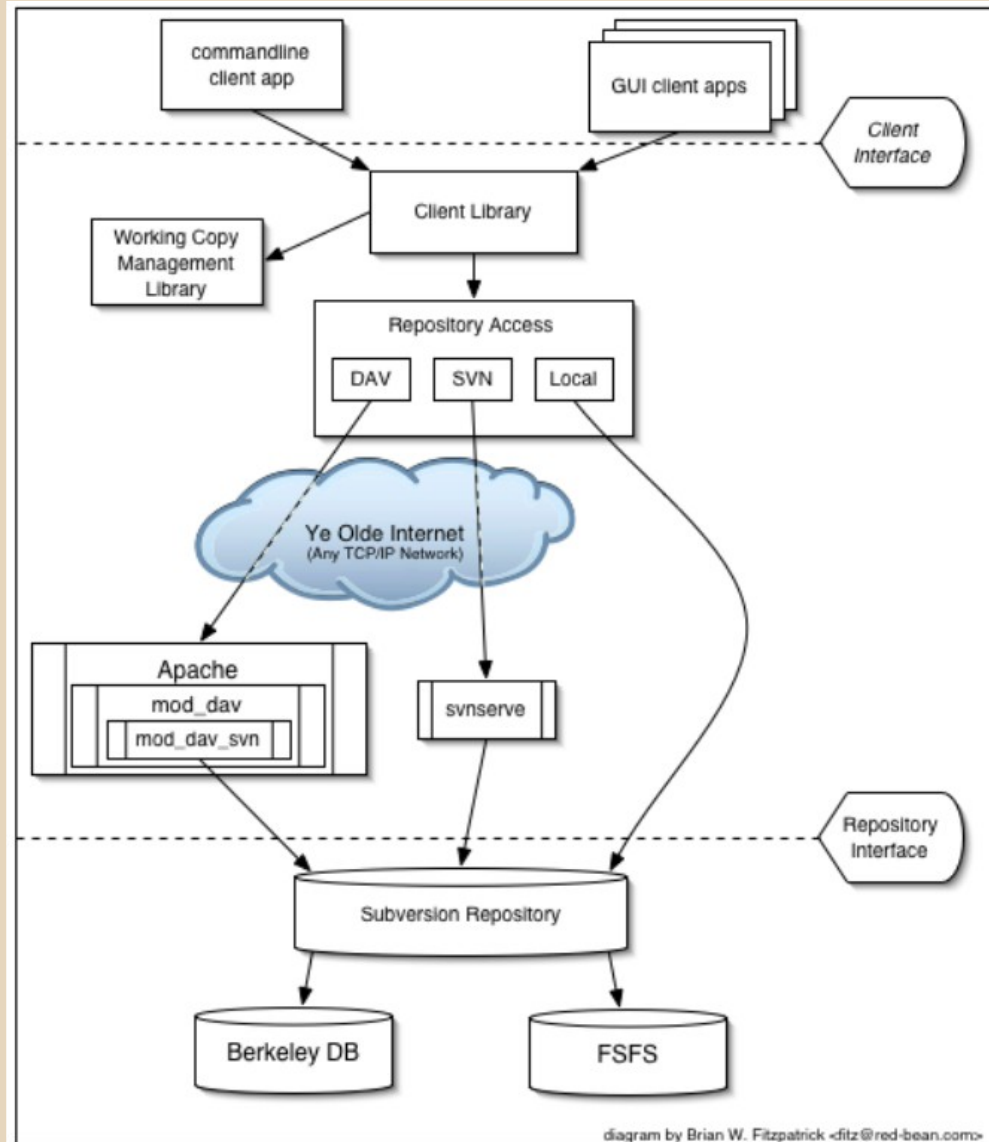
- Διατηρούν πλήρη αντίγραφα του πηγαίου κώδικα
- Συγκρίνουν τις εκδόσεις
- Συγχωνεύουν τις αλλαγές από διάφορες εκδόσεις
- Παρακολουθούν τις αλλαγών

Τι είναι το SVN

- Είναι μία δωρεάν / open source έκδοση ενός συστήματος ελέγχου (VCS) που διαχειρίζεται τα αρχεία, τους καταλόγους, και τις αλλαγές που γίνονται σε αυτούς με την πάροδο του χρόνου.
- Παρέχει ένα απλό μοντέλο: αριθμημένα στιγμιότυπα των δομών καταλόγου.

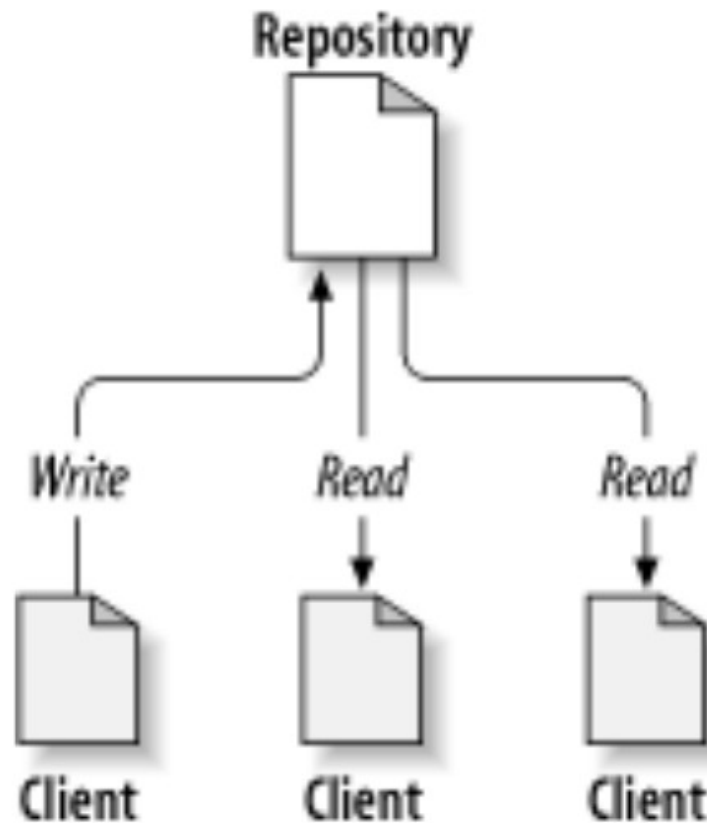
Αρχιτεκτονική SVN

πηγή:[1]



SVN model

Figure 1.1. A typical client/server system



πηγή:[1]

Εγκατάσταση SVN

- εγκατάσταση σε Ubuntu:

```
$ sudo apt get install subversion
```

- εγκατάσταση σε Fedora:

System->Administration -> Add/Remove Software ->
Search for “subversion”

Ορολογία

- Repository: Σετ εκδόσεων
- Revision: Ένα στιγμιότυπο ενός συγκεκριμένου καταλόγου των αρχείων
- Revision number: Ένας αύξων αριθμός που υποδηλώνει μια συγκεκριμένη αναθεώρηση
- Working copy: Ένας κατάλογος ή ένα αρχείο που περιέχει όλα τα αρχεία που έχουν ελεγχθεί μετά από την αναθεώρηση

Μια χρήσιμη ιδιότητα

- Δεν γίνεται να κρατηθούν πλήρη αντίγραφα των αρχείων.
- Αντίθετα, διατηρούνται οι διαφορές μεταξύ των εκδοχών: αν αλλάξετε λίγο τον κώδικα, η αναθεώρηση σας θα καταλαμβάνει λίγο χώρο.
- Η αντιγραφή ενός ολόκληρου αρχείου ή ενός καταλόγου στο repository είναι πολύ αποδοτική

Μερικοί βασικοί κανόνες

- Στα δημόσια repositories μπορείτε να έχετε πρόσβαση μέσω http και να δημιουργείτε αντίγραφα εργασίας στον υπολογιστή σας.
- Για να αποκτήσετε πρόσβαση στα αποθετήρια του Subversion σας, θα πρέπει να χρησιμοποιείτε SSH tunnels.
- Έχει δημιουργηθεί ένα ζευγάρι κλειδιών SSH για την πρώτη φορά που θα κάνετε σύνδεση.
- Παρακάτω θα θεωρούμε ένα χρήστη myUser και ένα server myServer που φιλοξενεί ένα repository της ομάδας myTeam.

Σύνοψη βασικών εντολών

command	description
<code>svn add <i>files</i></code>	schedule files to be added at next commit
<code>svn ci [<i>files</i>]</code>	commit / check in changed files
<code>svn co <i>repo</i></code>	check out
<code>svn help [<i>command</i>]</code>	get help info about a particular command
<code>svn import <i>directory repo</i></code>	adds a directory into repo as a project
<code>svn merge <i>source1 source2</i></code>	merge changes
<code>svn revert <i>files</i></code>	restore local copy to repo's version
<code>svn resolve <i>files</i></code>	resolve merging conflicts
<code>svn update [<i>files</i>]</code>	update local copy to latest version
others: blame, changelist, cleanup, diff, export, ls/mv/rm/mkdir, lock/unlock, log, propset	

πηγή: [2]

A day in the life of svn

- Αρχικά ενημερώνετε ένα αντίγραφο εργασίας (`svn update`)
- Ακολουθούν πιθανές αλλαγές (`svn add`, `svn delete`, `svn copy`, `svn move`)
- και αξιολόγηση αυτών (`svn status`, `svn diff`)
- διόρθωση λαθών (μπορεί να χρειαστεί να ξεκινήσετε από την αρχή: `svn revert`)
- προετοιμασία για να γίνουν `commit` οι αλλαγές (`svn update`, `svn resolve`)
- `commit` των αλλαγών (`svn commit`)
- επαναλάβετε πολλές φορές (πολύ καλή πρακτική: κάνετε `commit` μόλις οι αλλαγές στο πηγαίο κώδικα αποτελέσουν μία συνολικά λογική αλλαγή)

Δημιουργώντας ένα αντίγραφο εργασίας ενός repository

Για να λάβετε την τελευταία έκδοση ενός δημόσιου repository μέσω http:

```
svn checkout http://svn.example.com/svn/repo/trunk ή  
svn co http://svn.example.com/svn/repo/trunk
```

Για να λάβετε την τελευταία αναθεώρηση των project:

- `svn co svn + ssh :// myUser @ myHost / myTeam mydir`

Ή απλά έναν κατάλογο:

- `svn co svn + ssh :// myUser @ myHost / myTeam /proj1 mydir`

Μια συγκεκριμένη αναθεώρηση:

- `svn co-R100 svn + ssh :// myUser @ myHost / myTeam /proj1 OLD1`

Updating

- Για να λάβετε εκδόσεις των αρχείων από την πιο πρόσφατη αναθεώρηση, επιλέγετε τον κατάλογο που θέλετε να ενημερωθεί:

```
svn update
```

- Αυτό θα ενημερώσει τα αρχεία των αλλαγών, των προσθέσεων, των διαγραφών και των συγχωνεύσεων του repository στο τοπικό σας κατάλογο.

Προσθήκη, Διαγραφή και μετονομασία αρχείων και καταλόγων

- Όταν προσθέτετε ή αφαιρείτε ένα αρχείο ή έναν κατάλογο σε ένα αντίγραφο εργασίας, πρέπει να ενημερώνώνεται το SVN για το γεγονός:

```
svn add NEW-FILE
```

```
svn delete OLD-FILE-OR-DIR
```

```
svn move OLD-PLACE NEW-PLACE
```

- Αυτές οι εντολές δεν τροποποιούν το repository, μόνο τον τρέχοντα κατάλογο εργασίας.
- Οι αλλαγές πρέπει να γίνουν commit και να τροποποιηθεί το repository.

Προεπισκόπηση αλλαγών (1)

Για να δείτε τις αλλαγές που έχετε κάνει μέχρι τώρα στο αντίγραφο εργασίας σας:

```
$ svn status
? scratch.c
A stuff/loot
A stuff/loot/new.c
D stuff/old.c
M bar.c
```


Προεπισκόπηση αλλαγών (2)

? `item`

Το αρχείο, κατάλογος ή σύνδεσμος `item` δεν έχει προστεθεί στο svn.

A `item`

Το αρχείο, κατάλογος ή σύνδεσμος `item` έχει προστεθεί στο svn.

C `item`

Το αρχείο `item` έχει προκαλέσει σύγκρουση και πρέπει να επιλυθεί για να γίνει commit μαζί με τις υπόλοιπες αλλαγές.

D `item`

Το `item` θα διαγραφεί όταν γίνουν commit οι αλλαγές.

M `item`

Τα περιεχόμενα του `item` έχουν αλλάξει.

Συγκρίνοντας αναθεωρήσεις (1)

- Μια πολύ σημαντική ιδιότητα είναι η δυνατότητα σύγκρισης εκδόσεων και branches.
- Απλή περίπτωση: ποιες τοπικές αλλαγές έγιναν στον τρέχοντα κατάλογο εργασίας:

```
svn diff
```

- Σύγκριση του τρέχοντα καταλόγου εργασίας με την αναφεώρηση 9:

```
svn diff -r 9
```

- Διαφορά αναθεωρήσεων 9 και 10 με τον τρέχοντα κατάλογο

```
svn diff -r 9:10
```

Συγκρίνοντας αναθεωρήσεις (2)

```
$ svn diff
Index: bar.c
=====
--- bar.c (revision 3)
+++ bar.c (working copy)
@@ -1,7 +1,12 @@
+#include <sys/types.h>
+#include <sys/stat.h>
+#include <unistd.h>
+
+#include <stdio.h>

int main(void) {
- printf("Sixty-four slices of American Cheese...\n");
+ printf("Sixty-five slices of American Cheese...\n");
return 0;
}

Index: README
=====
--- README (revision 3)
+++ README (working copy)
@@ -193,3 +193,4 @@
+Note to self: pick up laundry.

Index: stuff/fish.c
=====
--- stuff/fish.c (revision 1)
+++ stuff/fish.c (working copy)
-Welcome to the file known as 'fish'.
-Information on fish will be here soon.

Index: stuff/things/blee.h
=====
```

πηγή:[1]

Επαναφορά παλιών εκδόσεων

- Πριν γίνει το commit μπορούμε να καταργήσουμε τις αλλαγές που έχουμε κάνει στο αρχείο.

- Η εντολή

```
$ svn revert FILE
```

αφαιρεί τις αλλαγές στο αρχείο.

- Κάνοντας revert σε μία τροποποίηση ή διαγραφή, επαναφέρουμε το αρχείο στη προηγούμενη κατάσταση.
- Κάνοντας revert σε μία προσθήκη αρχείου, αφαιρείται το αρχείο από το VCS χωρίς να διαγραφεί το αρχείο.

Κάνοντας commit τις αλλαγές

- Η εντολή `svn commit -m "Log message"` σε έναν κατάλογο εργασίας θα δημιουργήσει μια νέα αναθεώρηση του repository.
- Οι νέες αναθεωρήσεις διαφέρουν από την προηγούμενες στο περιεχόμενο του τρέχοντος καταλόγου, ο οποίος μπορεί να είναι μόνο μέρος ολόκληρου του δέντρου.
- Το μήνυμα πρέπει να είναι κατατοπιστικό. Αν απόφυγε το `-m`, θα καλέσουμε τον editor της αρεσκείας σας, που σας προτείνουμε, γιατί...
 - τα μηνύματα μητρώου πρέπει να είναι ακριβή και κατατοπιστικά
 - να εμφανίζονται δίπλα στο `svn log -v` και να μπορούν να βοηθήσουν τόσο εσάς όσο και άλλους που κοιτάζουν τις αλλαγές και καταλαβαίνουν τι συνέβη και έγιναν αλλαγές.

Commit

- Όταν δεν έχετε ενημερώσει το τοπικό σας αντίγραφο και δεν έχετε ελέγξει για τυχόν αλλαγές στο χώρο αποθήκευσης, το τοπικό σας αντίγραφο μπορεί να μην είναι ενημερωμένο:

```
$ svn commit --message "Add another rule"
```

Sending rules.txt

svn: Commit failed (details follow):

svn: Out of date: 'rules.txt' in transaction 'g'

- Στη συνέχεια, θα πρέπει να κάνετε `svn update` για να επιλυθούν οι πιθανές συγκρούσεις και ένα νέο commit.

Συγχώνευση

Συγχώνευση: δύο σετ αλλαγών εφαρμόζονται την ίδια στιγμή στα ίδια αρχεία. Συμβαίνει όταν δύο χρήστες δουλεύουν στην ίδια έκδοση αρχείων και κάνουν αλλαγές στα ίδια αρχεία.

- είτε και οι δύο κάνουν commit
- είτε ο ένας κάνει αλλαγές και τις κάνει commit και ο άλλος κάνει αλλαγές και μετά κάνει update.
- Για να συγχωνεύσετε τις αλλαγές μεταξύ δύο αναθεωρήσεων, R1 και R2, ενός αρχείου ή καταλόγου σε ένα αντίγραφο εργασίας, σημαίνει ότι θα κρατήσουμε τις αλλαγές που συνέβησαν μεταξύ των R1 και R2 και θα τις εφαρμόσουμε στο αντίγραφο εργασίας.
- Μετά τη συγχώνευση, όπως και στην ενημέρωση, πρέπει να επιλύσουμε τυχόν διαφορές (στη συνέχεια μπορούμε να κάνουμε commit τη συγχωνευμένη έκδοση).
- Για να συγχωνεύσετε τις αλλαγές στον τρέχοντα κατάλογο εργασίας (με την προϋπόθεση να είστε σε αυτόν τον κατάλογο):

```
svn merge SOURCE1@REV1 SOURCE2@REV2
```

όπου source1 και SOURCE2 είναι URLs (svn + ssh: ...) ή κατάλογοι εργασίας και REV1, REV2 είναι αριθμοί αναθεώρησης.

Συγκρούσεις

Σύγκρουση: συμβαίνει όταν το σύστημα δεν μπορεί να συνδέσει τις συγχωνευμένες αλλαγές.

- επίλυση σύγκρουσης: ο χρήστης παρεμβαίνει για να επιλύσει τη σύγκρουση. Πιθανοί τρόποι:
- κρατάει αλλαγές και από τις δύο εκδόσεις και τις ενώνει χειροκίνητα
- επιλέγει την έκδοση ενός χρήστη
- επαναφέρει τις αλλαγές και των δύο χρηστών

Όταν επιλύσουμε τη σύγκρουση, χρησιμοποιούμε:

`svn resolved FILE` ή `svn resolve --accept=working`
και στη συνέχεια κάνουμε `commit` τις αλλαγές

πηγή:[1]

```
...
Select: (p) postpone, (df) diff-full, (e) edit,
        (mc) mine-conflict, (tc) theirs-conflict,
        (s) show all options: df
--- .svn/text-base/sandwich.txt.svn-base      Tue Dec 11 21:33:57 2007
+++ .svn/tmp/tempfile.32.tmp                 Tue Dec 11 21:34:33 2007
@@ -1,5 @@
-Just buy a sandwich.
+<<<<<<<< .mine
+Go pick up a cheesesteak.
+=====
+Bring me a taco!
+>>>>>>> .r32
...
```


Προβολή Πληροφοριών

- Για να εμφανιστούν τα περιεχόμενα ενός καταλόγου ενός repository:

```
svn ls $ TEAMREPOS / tags
```

- Για να εμφανιστούν αναθεωρήσεις ενός αρχείου ή καταλόγου:

```
svn log FILE-OR-DIR
```

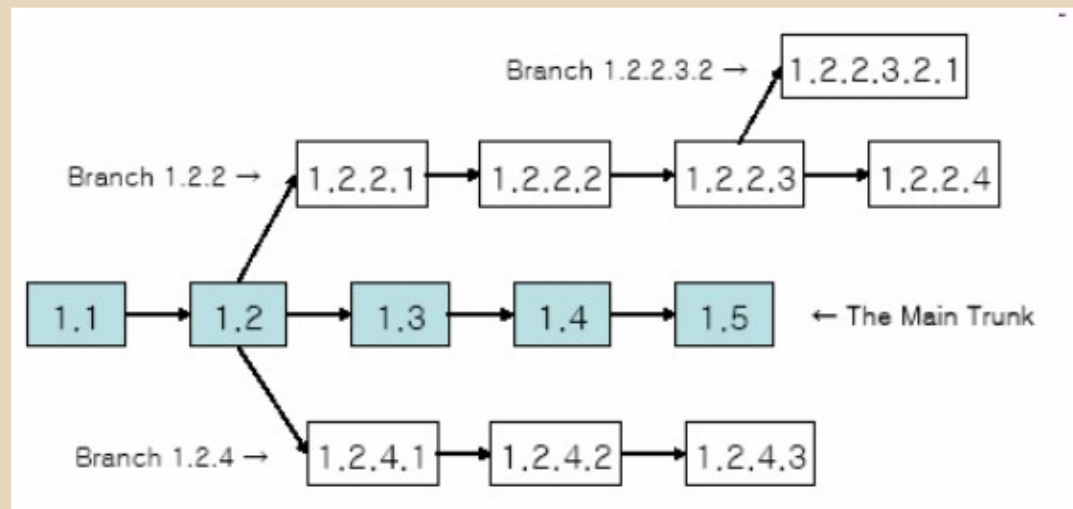
```
svn log -v FILE-OR-DIR # για λεπτομέρειες
```

Branches

Ένα δεύτερο αντίγραφο των αρχείων σε ένα repository

- τα δύο αντίγραφα μπορούν να αναπτυχθούν με διαφορετικούς τρόπους ανεξάρτητα
- δεδομένου του δικού τους αριθμού έκδοσης στο σύστημα ελέγχου έκδοσης
- τελικά συγχωνεύονται
- `trunk`: το κύριο αντίγραφο του κώδικα και όχι μέρος κάποιου fork

πηγή:[2]



Tags

Μια άλλη κοινή έννοια των VCS είναι τα tags. Πρόκειται για ένα "στιγμιότυπο" (snapshot) του προγράμματος σε μια χρονική στιγμή. Αυτή η έννοια είναι γνωστή καθώς κάθε αναθεώρηση του repository είναι ακριβώς αυτό, ένα στιγμιότυπο του συστήματος των αρχείων μετά από κάθε commit.

→ Για να δημιουργήσετε ένα tag του τρέχοντος αντιγράφου εργασίας σας (με την προϋπόθεση ότι βρίσκεστε σε αυτόν τον κατάλογο):

```
svn copy . http://svn.example.com/path/to/repository/tags/working-copy-seth
```

Μοντέλο δικτύου

Στην επικοινωνία του SVN-client με τον κεντρικό υπολογιστή:

- Ο SVN-client ασχολείται τον περισσότερο χρόνο με τη διεύθυνση των αντιφράφων εργασίας.
- Όταν χρειάζεται πληροφορίες από ένα απομακρυσμένο repository, κάνει ένα αίτημα προς το δίκτυο και ο διακομιστής απαντά με μια κατάλληλη απάντηση.
- Όταν ο server λάβει το αίτημα του πελάτη επιβεβαιώνει τη ταυτότητα του, και επιστρέφει την απάντηση.

Κλείνοντας

- Αν κάνετε commit εγκαίρως και συχνά, το σύστημα είναι αρκετά επιεικές.
- Μπορείτε να επαναφέρετε προηγούμενες καταστάσεις του κωδικά σας.
- Για να λειτουργήσει σωστά το σύστημα, θα πρέπει να κάνετε commit τακτικά και θα πρέπει να βεβαιωθείτε πως ότι χρειάζεστε είναι υπό τον έλεγχο έκδοσης (`svn status`).

Ένα άλλο VCS... GIT

Το git είναι ένα άλλο δημοφιλές σύστημα διαχείρισης εκδόσεων. Βασική διαφορά:

SVN: κεντρικό repository. Το repository του server είναι το κύριο με όλη την ιστορία των αλλαγών. Οι χρήστες κάνουν check out τοπικά αντίγραφα της τρέχουσας έκδοσης.

GIT: κατανεμημένο repository. Κάθε check out είναι ένα πλήρες repository με όλη την ιστορία του κώδικα. Προσφέρει ταχύτητα και redundancy(πλεονασμό). Οι διαδικασίες συγχώνευσης(merging) και διακλάδωσης(branching) είναι πολύ πιο συχνές.

πηγές

[1]<http://svnbook.red-bean.com/en/1.7/svn-book.pdf>

[2]:<http://courses.cs.washington.edu/courses/cse390a/>